YAMS : les systèmes d'automates à compteurs

Gilles Lesventes

INRIA

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP105
78153 Le Chesnay Cedex
France

Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 808

YAMS : LES SYSTEMES D'AUTOMATES A COMPTEURS

Gilles LESVENTES

MARS 1988

RR808

IRISA INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

PUBLICATION INTERNE N°392 - FEVRIER 1988 - 28 PAGES

YAMS : LES SYSTEMES D'AUTOMATES A COMPTEURS

Gilles LESVENTES
IRISA - Campus de Beaulieu
35042 RENNES CEDEX

Résumé :

Les systèmes d'addition de vecteurs représentent une classe assez large de modèles du parallélisme, parmi lesquels on trouve les réseaux de Pétri. Afin de garder la notion de processus et celle de leur composition, nous introduisons un nouveau modèle, YAMS , basé sur des automates finis communiquant au moyen de compteurs. Nous montrons la simulation réciproque de ce modèle et des réseaux de Pétri et l'égalité des langages des transitions reconnus par ces deux modèles.

YAMS : COUNTER AUTOMATA SYSTEMS

Abstract :

Vector addition systems represent a rather wide class of model of parallelism; among them are Petri nets. In order to keep the notions of processes and composition we introduce a new model , YAMS (for Yet Another Model of Synchronization), based on finite automata communicating by counters.
We show the mutual simulation of this model and Petri nets and the equality of their transition languages.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

Introduction :

Nous représentons un processus au moyen d'un système de transitions et d'un ensemble fini de variables. Le comportement d'un tel processus est décrit par un automate non déterministe dont les transitions sont divisées en deux classes: des actions dont le seul effet est de passer d'un état à un autre, et des opérations sur des variables appelées compteurs. Celles-ci sont, elles aussi, divisées en deux sortes : soit un incrément de 1 d'un seul compteur, soit un décrément de 1 également d'un compteur si celui-ci est positif.

Les processus sont naturellement faits pour être composés. Ceci se fait simplement par juxtaposition de processus liés entre eux par l'intersection de leurs ensembles de compteurs. Nous appelerons YAMS[1] un système d'automates à compteurs formé d'un ensemble d'automates d'états finis non déterministes et d'un vecteur de compteurs.

La définition d'un modèle amène toujours la comparaison avec d'autres modèles existants. Par le choix de processus asynchrones et par la manipulation de variables entières, il s'inscrit dans la classe des systèmes d'addition de vecteurs [KARP & MILLER] et comme pour ceux-ci nous faisons quelques hypothèses -ou restrictions- élémentaires: une opération sur une variable est effective en cela qu'elle modifie toujours et une seule fois la valeur de celle-ci ; l'ordre et les origines des opérations sur une variable ne peuvent être connus ; le mécanisme de rendez-vous entre processus n'existe pas et il n'y a pas de temps global ni même partiel.

Notons tout d'abord qu'un système d'automates à compteurs, même réduit à un seul automate serait identique à une machine de Minsky s'il possédait la faculté de tester la nullité d'un compteur et possèderait donc, avec deux compteurs au minimum, la puissance d'une machine de Turing [MINSKY].

Le modèle le plus proche des YAMS est, à notre connaissance, les systèmes d'addition de vecteurs avec états (VASS) de Hopcroft et Pansiot et on peut décrire le comportement des processus par un seul automate, produit de mélange de tous les automates, et ainsi voir les YAMS comme un cas particulier des VASS où l'on aurait limité à 1 ou -1 les opérations sur les compteurs [HOPCROFT & PANSIOT].

Nombre de modèles s'avèrent être équivalents quant à la puissance d'expression et nous montrons l'équivalence par simulation avec le plus cité d'entre eux, les réseaux de Pétri [PETRI]. Les YAMS sont en fait isomorphes à une sous-classe propre des réseaux de Pétri. Nous utiliserons de plus ceux-ci pour définir les classes de langages reconnus par les YAMS et montrons que les deux modèles reconnaissent les mêmes

[1] What ? Yet Another Model of Synchronization !

langages sur configurations bloquantes ou finales.

Nous pouvons citer enfin les réseaux à files de Finkel qui sont "des réseaux de Pétri où au lieu d'envoyer des marques (entiers) dans des places on envoie des mots dans des files" [FINKEL] et les automates finis communiquant par canaux FIFO [BRAND & ZAFIROPULO, ROSIER & YEN] . Ces modèles toutefois ont une puissance supérieure de part leur respect de l'ordre des évènements et perdent ainsi en facilité d'analyse.

Les propriétés classiques des modèles du parallélisme telles que la monotonie, la vivacité ou la finitude et les moyens de les déterminer ne sont pas décrites dans ce rapport mais peuvent néanmoins être obtenues de façon similaire aux réseaux de Pétri grâce à la bisimulation.

1 YAMS, le modèle :

1.1 Actions :

On note X l'alphabet des actions d'un système. Les lettres de cet alphabet ne sont pas signées.

1.2 Messages :

On note Y l'alphabet ordonné des messages. Il y a autant d'éléments dans Y qu'il y a de types différents de messages utilisés dans le système. Ses lettres peuvent être signées positivement traduisant l'émission d'un message, ou négativement traduisant la prise en compte d'un message.

1.3 Compteurs :

A chaque type de messages y, on fait correspondre un compteur c_y égal au nombre de messages de ce type émis et non encore pris en compte. .

1.4 Automate à compteurs :

Un automate à compteurs A_i est un automate d'états fini dont les arcs sont étiquetés par des lettres de X, $+Y=\{+y / y \in Y\}$, ou $-Y=\{-y / y \in Y\}$:

$$A_i = (Q_i , (X \cup +Y \cup -Y) , q_{i_0} , \delta_i)$$

où $\delta_i : Q_i \times (X \cup +Y \cup -Y) \rightarrow P(Q_i)$
q_{i_0} est l'état initial.

1.5 Système d'automates à compteurs :

Un système d'automates à compteurs est constitué d'un ou plusieurs automates à compteurs, sur les mêmes alphabets X et Y, et du vecteur des compteurs de Y:
Un système d'automate à compteurs, en "abrégé" YAMS, $\mathcal{A}$ est défini par:

$$\mathcal{A} = < X , Y , \{ A_i / i \in [n] \} , (E_I , S_I) >$$

où . n est le nombre d'automates dans le système

. $\forall\ i \in [n]$

$$A_i = (Q_i , (X \cup +Y \cup -Y) , q_{i_0} , \delta_i)$$

$$\delta_i : Q_i \times (X \cup +Y \cup -Y) \rightarrow P(Q_i)$$

. $E_I = (q_{1_0} , \ldots , q_{n_0})$ est la configuration initiale des états

. $S_I \in \mathbf{N}^{Card\ (Y)}$ est la configuration initiale des compteurs.

La relation de dérivation immédiate du système $\mathcal{A}$, notée $\underset{\mathcal{A}}{\Rightarrow}$, est définie comme suit:

$$\underset{\mathcal{A}}{\Rightarrow} : (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p . X_\varepsilon \rightarrow (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p \text{ où } p = \text{Card}(Y) \text{ et } X_\varepsilon = X \cup \{\varepsilon\}$$

$$((e_1, \ldots, e_n), (f_1, \ldots, f_p)) . a \underset{\mathcal{A}}{\Rightarrow} ((e_1', \ldots, e_n'), (f_1', \ldots, f_p'))$$

avec : si $a \in X$ et $h \in \delta_i(e_i, a)$

. $\forall j \in [n] \setminus \{i\} : e_j' = e_j$; $e_i' = h$; $\forall k \in [p] : f_k' = f_k$

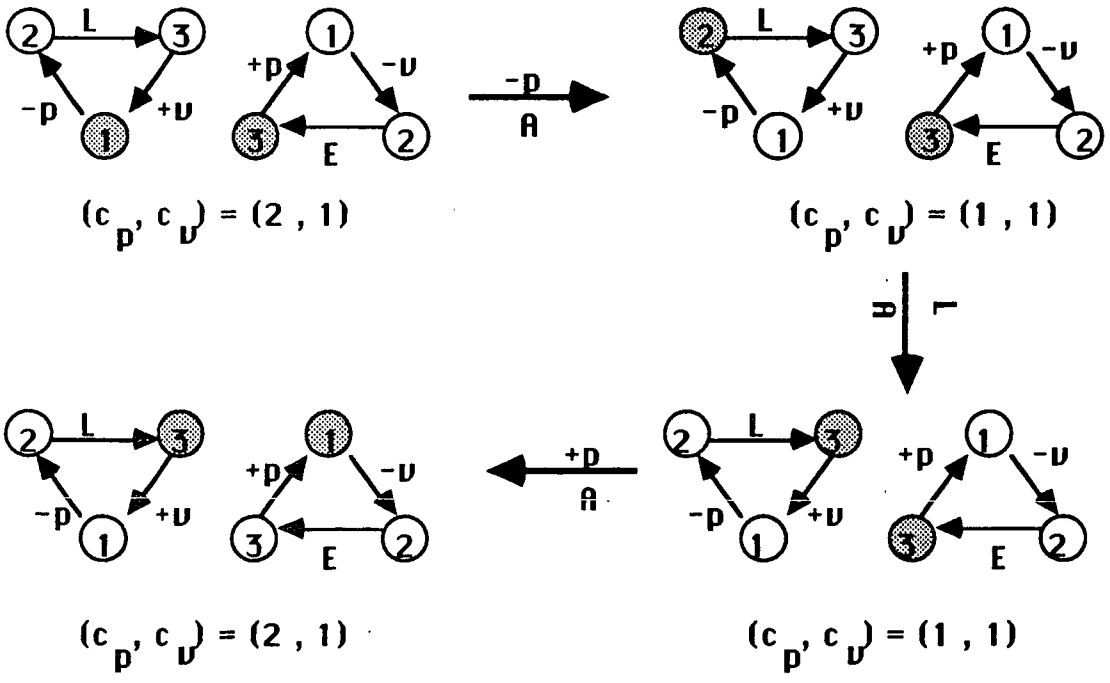
si $a = \varepsilon$ et $h \in \delta_i(e_i, -y_l)$, $l \in [p]$

. $\forall j \in [n] \setminus \{i\} : e_j' = e_j$; $e_i' = h$; $\forall k \in [p] \setminus \{l\} : f_k' = f_k$; $f_l' = f_l - 1$

si $a = \varepsilon$ et $h \in \delta_i(e_i, +y_l)$, $l \in [p]$

. $\forall j \in [n] \setminus \{i\} : e_j' = e_j$; $e_i' = h$; $\forall k \in [p] \setminus \{l\} : f_k' = f_k$; $f_l' = f_l + 1$

Exemple:



Soient $\underset{\mathcal{A}}{\rightarrow}$ la cloture de $\underset{\mathcal{A}}{\Rightarrow}$ par concaténation à droite sur X^* et $\overset{*}{\underset{\mathcal{A}}{\rightarrow}}$ la clôture réflexive et transitive de $\underset{\mathcal{A}}{\rightarrow}$.

1.6 Langages reconnus par un YAMS :

La définition des différents types de langages reconnus par un YAMS se fait selon différents ensembles de configurations atteintes; pour tout YAMS $\mathcal{A}$, de configuration initiale (E_I , S_I) :

Définition :

Le langage reconnu par le YAMS $\mathcal{A}$ est:

$$L(\mathcal{A}) = \{ u \in X^* / \exists (E,S) \in (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p : (E_I,S_I).u \xrightarrow[\mathcal{A}]{*} (E,S)\}.$$

Définition :

Un langage L est reconnu par le YAMS $\mathcal{A}$ sur configuration finale s'il existe un

ensemble de configurations F tel que :

$$LF(\mathcal{A}) = \{ u \in X^* / \exists (E,S) \in (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p : (E_I,S_I).u \xrightarrow[\mathcal{A}]{} (E,S) ; (E,S) \in F \}$$

Définition :

Un langage L est reconnu par le YAMS $\mathcal{A}$ sur configuration bloquante s'il existe un ensemble B de configurations tel que :

$$\forall (E,S) \in B , \forall a \in X_\varepsilon , \neg (\exists (E',S') \in (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p : (E,S) .a \underset{\mathcal{A}}{\Rightarrow} (E',S'))$$

$$\text{et } LF(\mathcal{A}) = \{ u \in X^* / \exists (E,S) \in (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p : (E_I,S_I).u \xrightarrow[\mathcal{A}]{*} (E,S) ; (E,S) \in B \}$$

2 YAMS et réseaux de Pétri :

remarques préliminaires :

Les réseaux de Pétri considérés sont ceux dont la fonction d'étiquetage σ est la plus générale c'est-à-dire ne nécessitant pas d'étiquettes distinctes pour chaque transition, et autorisant l'étiquetage par le mot vide ε.

La présentation des réseaux de Pétri utilisée est celle du [BRAMS].

2.1 Simulation:

2.1.1 Définitions :

Nous utilisons la notion d'homomorphismes entre systèmes de calculs décrite dans [KASAI, MILLER].

Nous redonnons les majeures définitions utilisées:

2.1.1.1 Système de calcul :

Définition :

Un système de calcul S consiste en :

- un ensemble D
- un élément x de D
- un ensemble fini Σ d'opérations
- une fonction $-$ de Σ dans l'ensemble des fonctions partielles de D dans D,

telle que pour tout $a \in \Sigma$, $\bar{a}$ soit une fonction partielle de D dans D.

La fonction $-$ est étendue à Σ^* par:

$\bar{\varepsilon}$ = identité

$\overline{\alpha\beta}(y) = \bar{\alpha} \circ \bar{\beta}(y) = \bar{\beta}(\bar{\alpha}(y))$; $\alpha,\beta \in \Sigma^*, y \in D$

Nous écrivons S= (Σ, D, x) au lieu de S= (Σ, D, x, $-$)

D est intuitivement vu comme l'ensemble des états du système de calcul, où un état inclut de l'information de contrôle aussi bien que des données pour la synchronisation. L'élément x est considéré comme l'état initial du système. L'éxécution d'une opération va créer un nouvel état , comme défini par la fonction $-$, et une séquence d'éxécutions d'opérations peut être vue comme une séquence de calculs du système.

2.1.1.2 Homomorphismes :

Définition :

Soient $S_1=(\Sigma_1,D_1,x_1)$ et $S_2=(\Sigma_2,D_2,x_2)$ des systèmes de calcul. Un homomorphisme h: $S_1 \rightarrow S_2$ consiste en :

- un homorphisme $\tau : \Sigma_1^* \rightarrow \Sigma_2^*$
- une injection $\rho : D_1 \rightarrow D_2$ qui satisfait les conditions suivantes:
  - $\rho(x_1) = x_2$
  - $\forall\ y,z \in R_{S_1}, \alpha \in \Sigma_1^* :\quad \overline{\alpha}(y) = z \Rightarrow \overline{\tau(\alpha)}\,(\rho(y)) = \rho(z)$

où R_{S_1} est l'ensemble des états accessibles de S_1 à partir de x_1 .

2.1.1.3 Simulation :

Définition :

Soient S_1 et S_2 deux systèmes de calcul. Supposons qu'il existe un homomorphisme h de S_1 dans S_2. Nous disons que S_1 est simulé par S_2 selon h.

2.1.2 Simulation d'un YAMS par un réseau de Pétri :

A tout YAMS $\mathcal{A}=(X,Y,\{A_i / i \in [n]\},(E_I,S_I))$ on va associer un réseau de Pétri $R_{\mathcal{A}}$ construit de la façon suivante:

$$. P = P_Q \cup P_Y \quad \text{avec } P_Q = \bigcup_{i=1}^{n} \{ p_{ij} / q_j \in Q_i \} \text{ et } P_Y = \{ p_y / y \in Q_i \}$$

$$. T = T_X \cup T_{+Y} \cup T_{-Y} \text{ avec}$$

$$T_X = \bigcup_{i=1}^{n} \{ t^i_{jkx} / q_k \in \delta_i (q_j, x) ; x \in X , (q_j,q_k) \in Q_i^2 \}$$

$$T_{+Y} = \bigcup_{i=1}^{n} \{ t^i_{jky+} / q_k \in \delta_i (q_j ,+y) ; y \in Y , (q_j,q_k) \in Q_i^2 \}$$

$$T_{-Y} = \bigcup_{i=1}^{n} \{ t^i_{jky-} / q_k \in \delta_i (q_j ,-y) ; y \in Y , (q_j,q_k) \in Q_i^2 \}$$

. $\forall\ t^i_{jkx} \in T_X$: $I(t^i_{jkx}) = \{p_{ij}\}$; $O(t^i_{jkx}) = \{p_{ik}\}$

$\forall\ t^i_{jky+} \in T_{+Y}$: $I(t^i_{jky+}) = \{p_{ij}\}$; $O(t^i_{jky+}) = \{p_{ik}, p_y\}$

$\forall\ t^i_{jky-} \in T_{-Y}$: $I(t^i_{jky-}) = \{p_{ij}, p_y\}$; $O(t^i_{jky-}) = \{p_{ik}\}$

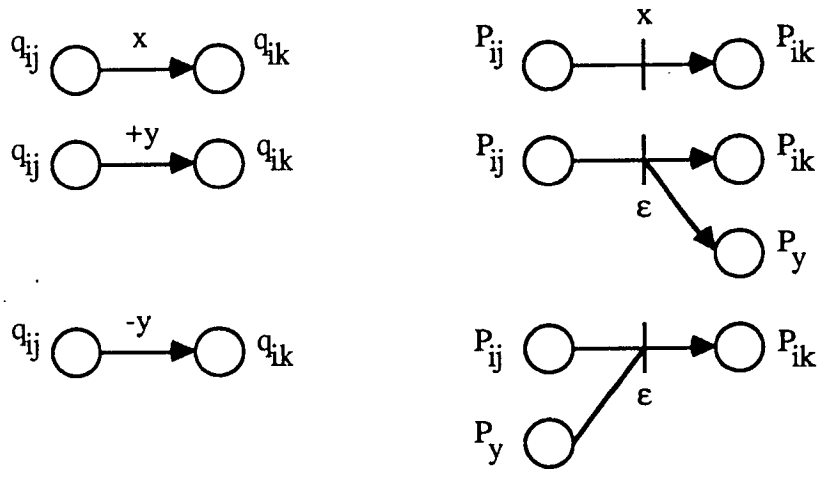
. la fonction d'étiquetage $\sigma : T \rightarrow X_\varepsilon$ est :

$\forall\ t^i_{jkx} \in T_X$: $\sigma(t^i_{jkx}) = x$

$\forall\ t^i_{jky+} \in T_{+Y}$: $\sigma(t^i_{jky+}) = \varepsilon$

$\forall\ t^i_{jky-} \in T_{-Y}$: $\sigma(t^i_{jky-}) = \varepsilon$

La construction est illustrée par la figure suivante:



YAMS Réseau de Pétri

. le marquage initial μ_I est défini par:

$\forall i \in [n] : (\dot{E}_I(i) = q_k \Rightarrow ((\forall j \in [\ Card(Q_i)\ ] \setminus \{k\} : \mu_I(p_{ij}) = 0) \wedge \mu_I(p_{ik}) = 1))$

$\forall r \in [p] : \mu_I(p_{y_r}) = S_I(r)$

Remarque :

Nous noterons $\bar{\beta}(u) = v$ par :

. $u\ (\beta > v$ dans le cas des réseaux de Pétri.

. $u \xrightarrow[\mathcal{A}]{\beta} v$ dans le cas des YAMS, tout en remarquant que

$\beta \in (X \cup +Y \cup -Y)^*$ et non pas à X^*.

Proposition 1 :
Quel que soit le YAMS $\mathcal{A}$ et son réseau de Pétri associé $R_{\mathcal{A}}$, il existe un homomorphisme H_1 de $S_{\mathcal{A}}$ dans $S_{R_{\mathcal{A}}}$ tel que $R_{\mathcal{A}}$ simule $\mathcal{A}$ selon H_1.

preuve :

Soit l'homomorphisme H_1 de $S_{\mathcal{A}} = (\bigcup_{i=1}^{n} \delta_i , \prod_{i=1}^{n} Q_i \times \mathbf{N}^p, (E_I , S_I))$ dans

$S_{R_{\mathcal{A}}} = (T, \mathbf{N}^{Card(P)}, \mu_I)$ construit ainsi :

Posons τ_0^i l'homomorphisme suivant, pour tout $i \in [n]$:

$\tau_0^i : \delta_i \rightarrow T$

$(q_j, x, q_k) \rightarrow t^i_{jkx}$

$(q_j, +y, q_k) \rightarrow t^i_{jky+}$

$(q_j, -y, q_k) \rightarrow t^i_{jky-}$

Soit $\tau_0 = \bigcup_{i=1}^{n} \tau_0^i$ et soit τ_1 l'extension de τ_0 aux séquences d'exécutions d'opérations
Soit l'injection $\rho_1 : \prod_{i=1}^{n} Q_i \times \mathbf{N}^p \rightarrow \mathbf{N}^{Card(P)}$

$(E,S) \longrightarrow (\mu(p_{11}), ..., \mu(p_{nm}), \mu(p_{y_1}), ..., \mu(p_{y_p}))$

où . $\forall i \in [n]: (E(i) = q_k) \Rightarrow ((\forall j \in [Card(Q_i)] \setminus \{q_k\}: \mu(p_{ij}) = 0) \wedge (\mu(p_{ik}) = 1))$

. $\forall r \in [p]: \mu(p_{y_r}) = S_I(r)$

Les conditions sur l'injection ρ_1, nécessaires pour que H_1 soit un homomorphisme entre deux systèmes de calcul, sont valides d'après le lemme suivant:

Lemme 1.1 :

La fonction ρ_1 vérifie les 2 propriétés suivantes:

a- $\rho_1 ((E_I,S_I)) = \mu_I$

b- $\forall$ (E_1,S_1), (E_2,S_2) tels que $\exists u,v \in (X \cup +Y \cup -Y)^*$ et

$$(E_I,S_I) \xrightarrow[\mathcal{A}]{u} (E_1,S_1)$$

$$(E_I,S_I) \xrightarrow[\mathcal{A}]{v} (E_2,S_2)$$

$\forall a \in (X \cup +Y \cup -Y)^*$

si $(E_1,S_1) \xrightarrow[\mathcal{A}]{a} (E_2,S_2)$ alors $\rho_1((E_1,S_1))$ $(\tau_1(a) > \rho_1((E_2,S_2))$

preuve :

a- évident

b- en raisonnant par récurrence sur $m = |a|$

.m=0: évident

.$\forall m > 0$: $a = a_1 \ldots a_m$

$$(E_1,S_1) \xrightarrow[\mathcal{A}]{a_1} \cdots \xrightarrow[\mathcal{A}]{a_{m-1}} (E_3,S_3) \xrightarrow[\mathcal{A}]{a_m} (E_2,S_2)$$

et $m_1 = \rho_1((E_1,S_1))$ $(\tau_1 (a_1 \ldots a_{m-1}) > \rho_1((E_3,S_3)) = \mu_3$

Rappelons que M (t > M' désigne le fait d'obtenir le marquage M' à partir de M en tirant la transition ou séquence de transitions t. (cf [BRAMS]).

3 cas sont à distinguer:

i - $a_m = (q_j, x, q_k)$, $x \in X$, $(q_j,q_k) \in Q_i^2$

· $\forall h \in [n] \setminus \{i\}$: $E_2(h) = E_3(h)$

. $E_2(i) = q_k$

. $S_2 = S_3$

On a bien $t^i_{jkx} = \tau_1(a_m)$

et $\forall p \in P \setminus \{p_{ij}, p_{ik}\}$: $\mu_2(p) = \mu_3(p)$

$\mu_2(p_{ij}) = \mu_3(p_{ij}) - 1$, soit 0

$\mu_2(p_{ik}) = \mu_3(p_{ik}) + 1$, soit 1

d'où $\mu_2 = \rho_1((E_2,S_2))$

ii - $a_m = (q_j, +y_l, q_k)$, $y_l \in Y$, $(q_j,q_k) \in Q_i^2$

· $\forall h \in [n] \setminus \{i\}$: $E_2(h) = E_3(h)$

. $E_2(i) = q_k$

· $\forall h \in [p] \setminus \{l\}$ $S_2(h) = S_3(h)$

. $S_2(l) = S_3(l) + 1$

On a bien $t^i_{jky+} = \tau_1(a_m)$

et $\forall p \in P \setminus \{p_{ij}, p_{ik}, p_{y_l}\}$: $\mu_2(p) = \mu_3(p)$

$\mu_2(p_{ij}) = \mu_3(p_{ij}) - 1$, soit 0

$\mu_2(p_{ik}) = \mu_3(p_{ik}) + 1$, soit 1

$\mu 2(p_{y_l}) = \mu 3(p_{y_l}) + 1$

d'où $\mu_2 = \rho_1((E_2,S_2))$

iii - $a_m = (q_j, -y_l, q_k)$, $y_l \in Y$, $(q_j,q_k) \in Q_i^2$

· $\forall h \in [n] \setminus \{i\}$: $E_2(h) = E_3(h)$

. $E_2(i) = q_k$

· $\forall h \in [p] \setminus \{l\}$ $S_2(h) = S_3(h)$

. $S_2(l) = S_3(l) - 1$

On a bien $t^i_{jky-} = \tau_1(a_m)$

et $\forall p \in P \setminus \{p_{ij}, p_{ik}, p_{y_l}\}$: $\mu_2(p) = \mu_3(p)$

$\mu_2(p_{ij}) = \mu_3(p_{ij}) - 1$, soit 0

$\mu_2(p_{ik}) = \mu_3(p_{ik}) + 1$, soit 1

$\mu 2(p_{y_l}) = \mu 3(p_{y_l}) - 1$

d'où $\mu_2 = \rho_1((E_2,S_2))$

Dans les 3 cas on a $\mu_2 = \rho_1((E_2,S_2))$ c'est-à-dire

$\rho_1((E_3,S_3))$ ($\tau_1(a_m) > \rho_1((E_2,S_2))$

et comme $\tau_1(a_1 \ldots a_{m-1}).\tau_1(a_m) = \tau_1(a_1 \ldots a_m) = \tau_1(a)$ on a bien

$\rho_1((E_1,S_1))$ ($\tau_1(a) > \rho_1((E_2,S_2))$.

(1.1)❑

(1)❑

L'homomorphisme τ_1, constituant de H_1, est une bijection. De plus on peut restreindre $\rho_1: (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p \rightarrow \mathbf{N}^{Card(P)}$ à $\rho_1': (\prod_{i=1}^{n} Q_i) \times \mathbf{N}^p \rightarrow \{0,1\}^n \times \mathbf{N}^p$ sans perte de généralité. Nous pouvons donc construire H_1^{-1}, constitué de τ_1^{-1} et de $\rho_1'^{-1}$.

De façon évidente nous obtenons la proposition suivante:

Proposition 2 :
Le YAMS $\mathcal{A}$ simule le réseau de Pétri $R_{\mathcal{A}}$ selon H_1^{-1}.

Définition :
Un homomorphisme h de S_1 dans S_2 préserve la longueur si $| \tau(a) | = 1$ pour tout $a \in \Sigma_1$.

Proposition 3 :
Les homomorphismes H_1 et H_1^{-1} préservent la longueur.

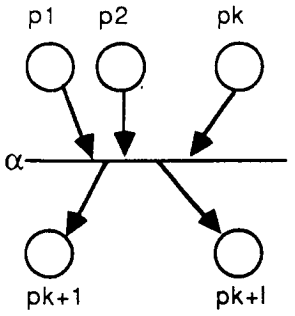
2.1.3 Simulation d'un réseau de Pétri par un YAMS :

A tout réseau de Pétri R=(P,T,I,O) et de marquage initial μ_I, on va associer un YAMS $\mathcal{A}_R$ construit de la façon suivante:

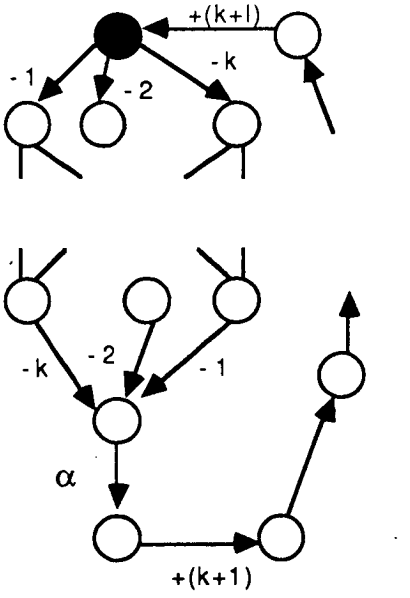
$$\mathcal{A}_R = (X, Y, \{A_i / t_i \in T\}, (E_I, S_I))$$

avec $X=\{\sigma(t) / t \in T\} \setminus \{\varepsilon\}$, et $Y=\{i / p_i \in P\}$ inclus dans $\mathbf{N}$.

Chaque transition t_i de R va être représentée par un automate A_i du système selon l'idée décrite par le schéma de la page suivante:



sera transcrit en



La construction de chaque automate A_i se décompose en trois phases:

1- Construisons un sous-automate $\overline{A_i}$ de A_i reconnaissant toutes les combinaisons des letttres (-a) telles que $a \in Y$ et $p_a \in I(t_i)$:
Notons Σ l'ensemble $\{-a / a \in Y \wedge p_a \in I(ti)\}$ et k le cardinal de Σ.
Définissons un ordre $<_\Sigma$ sur Σ tel que $-a <_\Sigma -b$ si $a <_{lexico} b$.

Soit $\overline{Q} = \{ q_{rs} / r \in [k+1], s \in [C_k^{r-1}] \}$ où r représente le fait que l'on ait reconnu (r-1) lettres de Y depuis l'état initial de $\overline{A_i}$, en noir sur le schéma, et s identifie l'une des combinaisons des (r-1) lettres prises parmi k.
Posons E_m tel que: $\forall m \geq 1\ E_m = \{ u \in \Sigma_m / \forall i \in [m+1], u(i) <_\Sigma u(i+1)\}$; on a $Card(E_m) = C_m^k$. E_m est l'ensemble des mots de m lettres de S dont les lettres sont en ordre croissant selon $<_\Sigma$. Soit f_m, la fonction qui donne à chacun de ces mots un rang suivant un ordre lexicographique $<_{\Sigma^*}$, extension de $<_\Sigma$ à Σ^*. Celle-ci permet de déterminer l'état où l'on se trouve après avoir lu m lettres, dans quelque ordre que ce soit:

$$f_m : E_m \rightarrow [C_m^k]$$

$$u \rightarrow f_m(u) = Card(\{v \in E_m / v <_{\Sigma^*} u\})$$

Nous pouvons définir maintenant la fonction de transition ν de ce sous-automate $\overline{A_i}$:

$\forall q_{rs} \in \overline{Q}$, $\forall -a \in \Sigma$:

$\nu(q_{rs},-a)$ est non défini si:

$(r=n+1)$ ou $(2 \leq r \leq n$ et $-a \notin \{u(t) / t \in [\,|u|\,]$ et $u=f_{r-1}^{-1}(s)\}\,)$

$\nu(q_{rs},-a) = q_{r+1s'}$ avec $s'=f_r(\, g(\, f_{r-1}^{-1}(s).-a\,)\,)$

où g est une fonction réordonnant le nouveau mot reconnu au rang r formé du mot reconnu au rang r-1 concaténé à -a :

$g: \{u \in \Sigma^t / \forall i,j \in [t]: u(i) \neq u(j)$ si $i \neq j\} \rightarrow E_t$

$u \rightarrow g(u)$ tel que $\{\, (g(u))(i) / i \in [\,|g(u)|\,]\,\} = \{\, u(i) / i \in [\,|u|\,]\,\}$

Le nombre d'états de ce sous-automate est : $\sum_{i=1}^{k+1} C_k^{i-1} = 2^k$.

Pour pouvoir construire A_i à partir de $\overline{A_i}$ nous devons procéder à une renumérotation φ de tout q_{rs} en q_j, $j \in [2k]$. On obtient ainsi les transitions de A_i suivantes: $q_j' \in \delta_i(q_j,-a) \Leftrightarrow \nu(\varphi^{-1}(q_j),-a) = \varphi^{-1}(q_j')$.

La –longue– construction de $\overline{A_i}$ permet de simuler les pré-conditions d'une transition du réseau de Pétri, c'est-à-dire $I(t_i)$.

2- On peut désormais reconnaitre –éventuellement– le label de cette transition: si $\sigma(t_i) \neq \varepsilon$ alors $\delta_i(q_{2k},\sigma(t_i)) = \{q_{2k+1}\}$ et on pose h=1 sinon on pose h=0.

3- La dernière partie de la construction de A_i permet de reconnaitre les post-conditions d'une transition d'un réseau c'est-à-dire $O(t_i)$. Quel que soit $a \in Y$ tel que $p_a \in O(t_i)$, il existe une transition $(q, +b_r, q')$ telle que $b_r=a$, parmi l'un des ensembles suivants:

$\{(q_m, +b_1, q_{m+1}), \ldots, (q_{m+l-2}, +b_{l-1}, q_{m+l-1}), (q_{m+l-1}, +b_{l-1}, q_1)\}$ si $l > 1$,

$\{(q_m, +b_1, q_1)\}$ si $l = 1$,

avec dans les deux cas : $l = Card(\, O(t_i)\,)$ et $m=2^k+h$.

En résumé: $\forall i$, $t_i \in T$: $Q_i = \{q_j / j \in [2^k+(l-1)+h]\}$ où $k = Card(I(t_i))$ et $h=0$ si $s(t_i)=e$, $h=1$ sinon.

Les configurations initiales de l'ensemble des automates A_i sont :

$E_I=(q_1, ..., q_1)$ pour les états, et $S_I=\mu_I$ pour le vecteur de compteurs.

Nous pouvons énoncer maintenant la proposition suivante:

Proposition 4 :

Quel que soit le réseau de Pétri R et son YAMS associé $\mathcal{A}_R$ par la construction précédente, il existe un homomorphisme H_2 de S_R dans $S_{\mathcal{A}_R}$ tel que $\mathcal{A}_R$ simule R selon H_2.

preuve :

Soit l'homomorphisme H_2 de $S_R=(T, N^{Card(P)}, \mu_I)$ dans

$S_{\mathcal{A}_R}= (\bigcup_{i=1}^{Card(T)} \delta_i, (\prod_{i=1}^{Card(T)} Q_i) \times \mathbf{N}^{Card(P)}, (E_I,S_I))$ construit ainsi:

Posons τ_0 l'homomorphisme suivant :

$$\tau_0 : T \rightarrow (\bigcup_{i=1}^{Card(T)} \delta_i)^*$$

$$t_j \rightarrow (q_1, -I_1, q_{I_1}) . (q_{I_1}, -I_2, q_{I_2}) . \dots . (q_{I_{h-1}}, -I_h, q_{2^h}) .$$

$$(q_{2^h}, \sigma(t_j), q_{2^h+1}) . (q_{2^h+1}, +O_1, q_{2^h+2}) . \dots . (q_{2^h+1}, +O_l, q_1) \text{ si } \sigma(t_j)\neq\varepsilon$$

$$t_j \rightarrow (q_1, -I_1, q_{I_1}) . \dots . (q_{I_{h-1}}, -I_h, q_{2^h})$$

$$(q_{2^h}, +O_1, q_{2^h+1}) . \dots . (q_{2^h+1}, +O_l, q_1) \text{ sinon.}$$

où dans les deux cas: $(I_1, ..., I_n)$ est tel que $\{p_{I1}, ..., p_{In}\}=I(t_j)$,

$(O_1, ..., O_n)$ est tel que $\{p_{O1}, ..., p_{On}\}=O(t_j)$ et

$\forall k \in [1,2^h+l]$ (ou $[1,2^h+l-1]$), $q_k \in Q_j$.

Soit t_2 l'extension de t_0 aux séquences de transitions.

Soit l'injection $\rho_2 : \mathbf{N}^{Card(P)} \rightarrow \prod_{i=1}^{Card(T)} Q_i \times \mathbf{N}^{Card(P)}$

$$\mu \rightarrow (q_1, ..., q_1) \times \mu$$

Lemme 4.1:

La fonction ρ_2 vérifie les deux propriétés suivantes:

a- $\rho_2(\mu_I)=(E_I,S_I)$

b- $\forall \mu_1,\mu_2$ tels que $\exists u,v \in T^*$: μ_I ($u > \mu_1$ et μ_I ($v > \mu_2$

$$\forall t \in T^*: \text{ si } \mu_1 \text{ (} t > \mu_2 \text{ alors } \rho_2(\mu_1) \overset{\tau_2(t)}{\underset{\mathcal{A}}{\rightarrow}} \rho_2(\mu_2)$$

preuve :

a- évident

b- par récurrence sur la longueur de t et d'après la construction des automates de $\mathcal{A}_R$ selon les transitions de T.

(4.1)❑

(4)❑

A toute transition de T, l'homomorphisme τ_2 fait correspondre une séquence de transitions de $\mathcal{A}_R$ de longueur supérieure ou égale à 2; H_2 ne préserve donc pas la longueur. On pourrait construire H_2^{-1} qui associe par τ_2^{-1} , à toute séquence de la forme $\tau_2(t)$, la transition t. La fonction inverse ρ_2^{-1} ne vérifierait pas la condition b-. En effet, il n'y aurait pas pour toute séquence de calcul de $\mathcal{A}$, en particulier tout préfixe de $\tau_2(t)$, quelle que soit t, de séquence associée dans R. On ne peut donc dire que R simule $\mathcal{A}_R$ selon H_2^{-1}.

2.2 Langages reconnus :

2.2.1 Types de langages d'un réseau de Pétri :

Peterson dans 'Petri nets and the modelling of systems' définit quatre types de langages associés à un réseau de Pétri; nous redonnons ses définitions [PETERSON]:

Définition :

Le langage $L_T(R)$ de type T associé à un réseau R de marquage initial μ_I est:

$$L_T(R)=\{u / u=\sigma(t) \wedge \mu_I (t>\mu \wedge \nexists t'\in T, \exists\mu': \mu (t'>\mu' \}.$$

Définition :

Le langage $L_L(R)$ de type L associé à un réseau R de marquage initial μ_I est tel qu'il existe un ensemble fini F de marquages finaux et:

$$L_L(R)=\{u / u=\sigma(t) \wedge \mu_I (t>\mu \wedge \mu \in F \}.$$

Définition :

Le langage $L_P(R)$ de type P associé à un réseau R de marquage initial μ_I est:

$$L_P(R)=\{u / u=\sigma(t) \wedge \mu_I (t>\mu \ , \ \mu \text{ quelconque} \}.$$

Définition :

Le langage $L_G(R)$ de type G associé à un réseau R de marquage initial μ_I est:

$$L_G(R)=\{u / u=\sigma(t) \wedge \mu_I (t>\mu \wedge \exists\mu'\in F : \mu \geq \mu' \}.$$

2.2.2 Equivalence de langages reconnus :

2.2.2.1 Equivalence $\mathcal{A} - R_{\mathcal{A}}$:

Proposition 5 :

Quel que soit le YAMS $\mathcal{A}$, le langage reconnu par celui-ci sur configuration bloquante est égal au langage de type T du réseau de Pétri $R_{\mathcal{A}}$ (associé à $\mathcal{A}$ par la construction du paragraphe 2.1.2).

preuve :

Définition :

Le label l d'une séquence de transitions de A est défini par:

$l ((q_j, x, q_k)) = x$

$l ((q_j, +y, q_k)) = +y$

$l ((q_j, -y, q_k)) = -y$

$l (a.b) = l (a).l (b)$

Lemme 5.1 :
Le langage $L_B(\mathcal{A})$ reconnu par $\mathcal{A}$ sur configuration bloquante est inclus dans le langage $L_T(R_{\mathcal{A}})$ de type T de $R_{\mathcal{A}}$.

preuve :
$R_{\mathcal{A}}$ simule $\mathcal{A}$ selon H_1, et H_1 préserve la longueur c'est-à-dire que toute transition de $\mathcal{A}$ est simulée par une transition de $R_{\mathcal{A}}$.
Soit ψ la projection de $(X \cup +Y \cup -Y)^*$ sur X^*.

$$L_B(\mathcal{A}) = \{ \psi(l(a)) / (E_I,S_I) \xrightarrow[\mathcal{A}]{a} (E,S),\ (E,S) \text{ bloquante}\}$$

On a aussi: $\forall a \in (\bigcup_{i=1}^{n} \delta_i)\ \ \psi(l(a))=\sigma(\tau_1(a))$.

De plus on a le résultat suivant:

Lemme 5.2 :

$$\forall a \in (\bigcup_{i=1}^{n} \delta_i) \text{ tel que } \psi(l(a)) \in L_B(\mathcal{A}): \mu_I\ (\tau_1(a) > \mu \text{ et } \mu \text{ bloquant.}$$

preuve :

$$\psi(l(a)) \in L_B(\mathcal{A}) \Rightarrow\ ((E_I,S_I) \xrightarrow[\mathcal{A}]{a} (E,S) \text{ et } (E,S) \text{ bloquante })$$

remarque: (E,S) bloquante $\Rightarrow$

$$\forall i \in [n]:\ \nexists\ q_k \in Q_i : (\ (\exists x \in X: q_k \in \delta_i(E(i),x) \vee (\exists y \in Y: q_k \in \delta_i(E(i),+y))\ )$$

$$\forall i \in [n], \forall k \in Q_i, \forall l \in [p]: (\ q_k \in \delta_i(E(i),-y) \Rightarrow S(l)=0\ ).$$

Soit $t^i_{jky-} = \tau_1(E(i), -y_l, q_k)$:

$$(\ I(t^i_{jky-}) = \{p_{ij}, p_y\} \text{ et } \mu(p_y)=0\ ) \Rightarrow t^i_{jky-} \text{ non tirable.}$$

(5.2)❑

Donc on a $\forall a \in (\bigcup_{i=1}^{n} \delta_i) : \sigma(\tau_1(a)) \in L_T(R_{\mathcal{A}})$.

(5.1)❑

Lemme 5.3 :
Le langage $L_T(R_{\mathcal{A}})$ de type T de $R_{\mathcal{A}}$ est inclus dans le langage $L_B(\mathcal{A})$ reconnu par $\mathcal{A}$ sur configuration bloquante .
preuve :
$R_{\mathcal{A}}$ simule $\mathcal{A}$ selon H_1^{-1}, et H_1^{-1} préserve la longueur c'est-à-dire que toute transition de $R_{\mathcal{A}}$ est simulée par une transition de $\mathcal{A}$.

$$L_T(R_{\mathcal{A}}) = \{u / u=\sigma(t) \wedge \mu_I(t > \mu \wedge \nexists t' \in T, \nexists \mu: \mu(t' > \mu')\}$$

On a aussi: $\forall t \in T^*: \sigma(t) = \psi(\tau_1^{-1}(t))$, et de plus le résultat suivant:

Lemme 5.4 :

$$\forall t \in T^* \text{ tel que } \sigma(t) \in L_T(R_A): (E_I,S_I) \xrightarrow[\mathcal{A}]{\tau_1^{-1}(t)} (E,S) \text{ et } (E,S) \text{ bloquante}$$

preuve :

$(\sigma(t) \in L_T(R_{\mathcal{A}}) \Rightarrow \mu_I(t > \mu$ et μ bloquant); c'est-à-dire $\nexists t' \in T, \nexists \mu': \mu(t' > \mu')$
On remarque l'implication suivante:

$$(\exists \mu \text{ tel que } \nexists t' \in T, \nexists \mu': \mu(t' > \mu') \Rightarrow$$
$$(\forall t \in T_X \cup T_{+Y}: I(t)=\{p\} \text{ avec } p \in P_Q \text{ et } \mu(p)=0)$$

D'où l'on déduit:

$$\forall i \in [n], \forall j \in \text{Card}(Q_i) \text{ tels que } \mu(p_{ij})>0 :$$

$$\forall l \in [p], \forall k \in \text{Card}(Q_i) : (t^i_{jky_l^-} \in T \Rightarrow \mu(p_{y_l})=0)$$

Soit $(q_j, -y_l, q_k) = t_1^{-1}(t^i_{jky_l^-})$: $(E(i)=q_j \wedge S(l)=0) \Rightarrow$ transition non tirable

(5.4)❑

Donc on a : $\forall t \in T^* \ \psi(\tau_1^{-1}(t)) \in L_B(\mathcal{A})$.

(5.3)❑
(5)❑

Proposition 6 :
Quel que soit le YAMS $\mathcal{A}$, le langage reconnu par celui-ci sur configurations finales est égal au langage de type L du réseau de Pétri $R_{\mathcal{A}}$ (associé à $\mathcal{A}$ par la construction du paragraphe 2.1.2).

preuve :

Soit $F_{\mathcal{A}} \subset \prod_{i=1}^{n} Q_i \times \mathbb{N}^p$, l'ensemble fini des configurations finales de $\mathcal{A}$, $F_{R_{\mathcal{A}}}$ l'ensemble des marquages finaux de $R_{\mathcal{A}}$ sera donné par:

$F_{R_{\mathcal{A}}} = \{ \rho_1((E,S)) / (E,S) \in F_{\mathcal{A}} \}$

Puisque $R_{\mathcal{A}}$ simule $\mathcal{A}$ selon H_1 et que $\mathcal{A}$ simule $R_{\mathcal{A}}$ selon H_1^{-1}, les deux préservant la longueur et d'après la construction de $F_{R_{\mathcal{A}}}$:

$\forall a \in (\bigcup_{i=1}^{n} \delta_i)$ $\psi(l(a)) = \sigma(\tau_1(a))$ implique l'égalité des deux langages.

(6)❑

Tout aussi évidemment nous obtenons, sans en fournir la preuve, la proposition suivante:

Proposition 7 :

Quel que soit le YAMS $\mathcal{A}$, le langage reconnu par celui-ci est égal au langage de type P du réseau de Pétri $R_{\mathcal{A}}$ (associé à $\mathcal{A}$ par la construction du paragraphe 2.1.2).

❑

2.2.2.2 Equivalence $R - \mathcal{A}_R$:

Proposition 8 :

Quel que soit le réseau de Pétri R, le langage de type L reconnu par celui-ci est égal au langage reconnu sur configurations finales par le YAMS $\mathcal{A}_R$ (associé à R par la construction du paragraphe 2.1.3).

preuve :

$\mathcal{A}_R$ simule R selon H_2. A toute transition t de R, σ_2 associe une séquence de transitions de $\mathcal{A}_R$, simulant les pré-conditions, l'étiquette et les post-conditions de t.

On a donc $\forall t \in T^* : \sigma(t) = \psi(\tau_2(t))$.

Soit F_R l'ensemble des marquages finaux de R, $F_{\mathcal{A}_R}$ l'ensemble des configurations finales de A_R sera donné par:

$F_{\mathcal{A}_R} = \{\rho_2(\mu) / \mu \in F_R \}$

lemme 8.1 :
Le langage $L_L(R)$ est inclus dans le langage $L_F(\mathcal{A}_R)$.

preuve :
De façon évidente d'après la construction de $F_{\mathcal{A}_R}$ et d'après le fait que $\mathcal{A}_R$ simule R selon H_2.

(8.1)□

lemme 8.2 :
Le langage $L_F(\mathcal{A}_R)$ est inclus dans le langage $L_L(R)$.

preuve :
En règle générale, l'inverse de H_2 n'est pas un homomorphisme entre R et $\mathcal{A}_R$; en cela qu'il ne peut simuler les séquences de transitions de $\mathcal{A}_R$ qui ne soient pas, à un réordonnancement près, une suite de suites de transitions de la forme $\tau_2(t)$, où t est une transition de R.
Dans ce cas de reconnaissance sur configurations finales et d'après la construction de $F_{\mathcal{A}_R}$, nous sommes dans la situation où toute séquence S de $\mathcal{A}_R$ reconnaissant un mot l(S) et atteignant une configuration finale est un mélange de séquences S_i d'une des formes suivantes:

$S_i = ((q_1, -\beta_{i1}, q_2), \dots , (q_j, -\beta_{ik}, q_{2k}), (q_{2k}, \alpha_i, q_{2k+1}),$
$(q_{2k+1}, +\gamma_{i1}, q_{2k+2}), \dots , (q_{2k+1}, +\gamma_{il}, q_1)$

ou

$S_i = ((q_1, -\beta_{i1}, q_2), \dots , (q_j, -\beta_{ik}, q_{2k}),$
$(q_{2k}, +\gamma_{i1}, q_{2k+1}), \dots , (q_{2k+1-1}, +\gamma_{il}, q_1)$

Notons Z l'ensemble de ses séquences S_i.

Lemme 8.3 :
Pour toute séquence S, mélange de n séquences S_i, il existe une séquence S_{CONC}, concaténation des n séquences S_i composant S reconnaissant le même mot et atteignant la même configuration.

preuve :
Soit t la transition (q_j, x, q_k), $(q_j, q_k) \in Q_i^2$ et $(x=\alpha_i, \alpha_i \in X$ ou $x=+\gamma_{i1}, \gamma_{i1} \in +Y)$ telle que :
si S = U t V, U et V étant deux séquences de transitions
alors $\forall t' \in U$: t' est de la forme $(q_r, -\beta, q_s)$, $(q_r, q_s) \in Q_h^2$, $h \in Card(T)$, $\beta \in Y$

Soit alors $S_C = S_i \,.\, \psi_i(U) \,.\psi_i(V) = S_i \,.\, \psi_i(S)$, où S_i est la séquence contenant t $-(S_i \in Z)-$ et ψ_i la projection de l'ensemble des transitions de Z sur l'ensemble des

transitions de $(Z \setminus \{S_i\})$.
Plusieurs remarques sont à faire:

- $\psi_i(S)$ est donc le mélange de (n-1) séquences.
- U est une séquence de transitions du type $(q_r, -\beta, q_s)$. Si, à partir d'une configuration, toute transition de U est franchissable, alors si U' est une séquence des transitions de U dans un ordre quelconque, toute transition de U' est franchissable, à partir de la même configuration.
- Si une transition est franchissable pour un vecteur de compteurs donné, elle le sera aussi pour un vecteur supérieur ou égal. Autrement dit, une transition de type $(q_r, +\gamma, q_s)$ ou du type (q_r, α, q_s) n'infirme aucunement la franchissabilité d'une transition quelconque.

De ces remarques nous voyons qu'une séquence S, mélange de n séquences de Z peut être ramenée à une séquence S_C, concaténation d'une séquence de Z et d'une séquence mélange de (n-1) séquences de Z. D'où, par récurrence sur n, on construit la séquence S_{CONC} et déduit le lemme.

(8.3)❑

A partir de la séquence S_{CONC}, on peut construire une séquence S' de transitions du réseau de Pétri R de la façon suivante: à chaque séquence de transitions S_i de Z on peut associer la transition t_i de T d'après la construction de $\mathcal{A}_R$. On voit aisément que quelle que soit S_{CONC}, la séquence S' correspondant reconnait le même mot et atteint un marquage équivalent à la configuration atteinte par S_{CONC} –cas où l'inverse de H_2 peut être appliqué– donc appartenant à F_R puisque la configuration atteinte par S appartient à $F_{\mathcal{A}_R}$.

(8.2)❑
(8)❑

2.2.2.3 Equivalence R - $\mathcal{B}_R$:

De la même façon que dans le paragraphe 2.2.2.1 – équivalence $\mathcal{A}$ - $R_{\mathcal{A}}$ –, nous aimerions pouvoir utiliser la même construction, donc le même YAMS $\mathcal{A}_R$, pour établir qu'il reconnait sur configuration bloquante le langage de type T associé à R. Mais on voit aisément que $\mathcal{A}_R$ peut reconnaître par configuration bloquante des préfixes du langage de type T associé à R. Informellement on peut dire que deux automates A_i et A_j de $\mathcal{A}_R$ peuvent "s'interbloquer" dans leurs séquences de transitions de $\overline{A_i}$ et de $\overline{A_j}$.

Pour pallier à ces problèmes d'interblocage entre automates d'un même YAMS on va associer à tout réseau de Pétri R, un YAMS $\mathcal{B}_R$ composé d'un unique –gros– automate en s'inspirant de la construction de $\mathcal{A}_R$:

De façon similaire au sous-automate $\overline{A_i}$ de chacun des A_i de $\mathcal{A}_R$, qui reconnait toutes les combinaisons correspondant aux pré-conditions d'une transition t_i de R, nous construisons un seul sous-automate $\overline{B}$ reconnaissant toutes les combinaisons correspondant aux pré-conditions de l'ensemble des transitions de R.

A chaque sommet de $\overline{B}$ sont associées les séquences de transitions, semblables à la fin des séquences S_i du YAMS $\mathcal{A}_R$ c'est-à-dire ($(q_{2k}, \alpha_i, q_{2k+1})$, $(q_{2k+1}, +\gamma_{i1}, q_{2k+2})$,..., $(q_{2k+l}, +\gamma_{il}, q_l)$), qui sont tirables pour ce sommet.

Définition :
Une séquence S est dite tirable pour un sommet qr si et seulement si, si t est la transition de R correspondant à S, quelle que soit p_a de I(t), -a appartient à l'ensemble des lettres de l'étiquette du chemin allant de q_1 à q_r.

Les séquences associées à un sommet sont identiques à la fin de séquence S_i décrite ci-dessus hormis la dernière transition; celle-ci est de la forme $(q_s, +\gamma_{il}, q_v)$ avec $\mathfrak{E}_v = \mathfrak{E}_s \setminus \{-a / p_a \in I(t)\}$ où $\mathfrak{E}_k$ est l'ensemble des lettres de l'étiquette d'un chemin allant de q_1 à q_k.
La configuration initiale (E_I,S_I) de $\mathfrak{B}_R$ est: $E_I = (q_1)$ pour l'état initial, $S_I = \mu_I$ pour le vecteur de compteurs.

Lemme 9.1 :
Pour toute séquence S de transitions de R reconnaissant un mot $\sigma(S)$ et atteignant un marquage μ, il existe une séquence S' de transitions de $\mathfrak{B}_R$ reconnaissant $l(S')=\sigma(S)$, et atteignant une configuration (E,S) telle que : $E=(q_1)$ et $S=\mu$.

preuve :
Par récurrence sur m=|S|
.m=0 : évident
.$\forall m>0$ $S=S_{m-1} . t_m$ avec $l(S_{m-1})=\sigma(S'_{m-1})$
D'après la construction de $\mathfrak{B}_R$, il existe un chemin partant de q_1 dont l'étiquette est une combinaison des pré-conditions de t_m. Soit q_r le sommet atteint à l'issue de ce chemin; t_m est tirable pour q_r et la séquence associée à q_r , et t_m est composée d'une action $\alpha=\sigma(t_m)$ et d'actions $(+\gamma_i, \gamma_i \in Y$ tel que $p\gamma_i \in O(t_m))$ et telle que la dernière transition est de la forme $(q_v, +\gamma_l, q_1)$.
On a donc $l(S') = l(S'_{m-1}).\alpha = \sigma(S_{m-1}).\sigma(t_m) = \sigma(S)$ et de plus $E=(q_1)$ et $S=\mu$.

De plus si S atteint un marquage μ bloquant, S' atteint une configuration (E,S) telle qu'il n'existe pas de chemin de q_1 vers un état quelconque q_k de $\overline{B}$ tel qu'une séquence soit tirable en q_k – puisque $\forall t \in T, \exists p \in I(t): \mu(p)=0$, sur R – .

Donc tous les comportements à partir de (E,S) mènent en une configuration bloquante sans avoir modifié le mot reconnu par $\mathfrak{B}_R$.

(9.1)□

On déduit donc le lemme suivant.

Lemme 9.2 :
Le langage $L_T(R)$ de type T est inclus dans le langage $L_B(\mathfrak{B}_R)$ reconnu par $\mathfrak{B}_R$ sur configuration bloquante.

La réciproque est établie grâce au lemme suivant.

Lemme 9.3 :
Pour toute séquence S de transitions de $\mathfrak{B}_R$ reconnaissant un mot l(S), il existe une séquence S' de transitions de R reconnaissant $\sigma(S')=l(S)$.

preuve :
S peut être vu comme le facteur gauche d'une séquence, concaténation de séquences S_i ayant la forme suivante : une suite de transitions $-\beta$, $\beta \in Y$, mélange de pré-conditions de différentes transitions de R, suivie -éventuellement- d'une transition α, ($\alpha \in X$, $\alpha=\sigma(t)$, $t \in T$), puis d'une suite de transitions $+\gamma$, $\gamma \in Y$, post-conditions de la transition t en question.

S' sera la séquence des transitions t des séquences S_i de S et reconnait donc le même mot que S en atteignant un marquage μ tel que : ((E,SP) est la configuration atteinte par $\mathfrak{B}_R$ à l'issue de S)

si $E = (q_r)$ et

- si $q_r \in \overline{B}$: soit V l'ensemble des $-\beta$ du chemin entre q_1 et q_r

  $\forall i \in Y : -i \in V \Rightarrow \mu(p_i)=SP(i)$

- si $q_r \notin \overline{B}$ et q_r appartient à un chemin entre un état de $\overline{B}$ et q_1:

  soit U l'ensemble des $+\gamma$ du chemin entre q_r et q_1

  $\forall i \in Y : +i \in U \Rightarrow \mu(p_i)>SP(i)$

De plus si $\mathfrak{B}_R$ est dans une configuration bloquante, c'est-à-dire que l'état courant est un état de $\overline{B}$ d'où aucune transition $-\beta$, $\beta \in Y$, n'est éxécutable et d'où aucune séquence n'est tirable; alors cela revient à dire que le marquage μ correspondant est bloquant.

(9.3)❑

On en déduit le lemme suivant.

Lemme 9.4 :
Le langage reconnu par $\mathfrak{B}_R$ sur configuration bloquante est inclus dans le langage de type T associé à R.

Des lemmes précédents on obtient la proposition suivante.

Proposition 9 :
Quel que soit le réseau de Pétri R, le langage de type T reconnu par celui-ci est égal au langage reconnu sur configuration bloquante par le YAMS $\mathfrak{B}_R$.

2.2.3 Conclusion sur les langages reconnus :

Les YAMS et les réseaux de Pétri reconnaissent les mêmes classes de langages que ce soit sur configurations finales (respectivement marquages finaux pour les réseaux de Pétri) ou sur configuration bloquante (respectivement marquage bloquant).
Les langages reconnus par les YAMS, c'est-à-dire sans considération de configuration bloquante ou finale, sont identiques aux langages de type P des réseaux de Pétri.Pour l'établir, il suffit de faire les équivalences R - $\mathfrak{B}_R$ et $\mathfrak{B}_R$ - R.

REFERENCES :

D.BRAND & P. ZAFIROPULO :
"On communicating finite-state machines" J.A.C.M. vol n° 2, 1983

A.FINKEL :
"Structuration des systèmes de transitions - Applications au contrôle du parallélisme par files FIFO" , Thèse d'état , ORSAY, 1987

M. HACK :
"Decidability questions for Petri nets", PhD dissertation, Dept of electrical engineering, Massachussets Institute of Technology, 1975

J. HOPCROFT & J.J. PANSIOT :
"On the reachability problem for 5-dimensional vector addition systems", Therotical Computer Science 8, 1979

R. KARP & R. MILLER :
"Parallel program shemata", Journal of Computer and System Science Vol 3, number 4, 1969

T. KASAI & R. MILLER :
"Homomorphisms between models of parallel computation", Research report IBM RC 7796, 1979

M. MINSKY :
"Computation: finite and infinite machines", Prentice-Hall, 1967

J.L. PETERSON :
"Petri nets theory and the modeling of systems", Prentice-Hall, 1981

C. PETRI :
"Fundamentals of a theory of asynchronous information flow", Proceedings of the 1962 IFIP Congress, North-Holland, 1962

"Kommunikation mit Automaten", PhD dissertation, University of Bonn, 1962

L.E. ROSIER & H.C. YEN :
"Boundedness, empty channel detection, and synchronisation for communicating finite automata", STACS 1985, L.N.C.S. n° 210

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique